**Animal Checker 设计文档**

**程序结构设计与分析**

1. 读入和初始化地形地图、动物地图的方法：
2. 先用两个一维的string数组接下两个文件都出来的东西；
3. 再用两个二维的char数组分别接收两个string数组上面的字符；
4. 启用第一个子方法，将两个char数组导入，经过判断后初始化一个二维的string数组；

2． 只有一个类；

3. 关键变量及其作用：

（1） atlas： 是一个二维的string数组，用于直接储存要打印出来的地图信息；

（2） allAtlas : 是一个string的三维数组，用于储存所有的棋盘信息；

（3） 用于undo 和 redo 的两个int 数据：currentStep 和initialStep

CurrentStep 用于表示当前步数；initialStep用于表示下过之后的原始的这一步，是悔棋的起点和redo的重点；

1. 用于判断输入输出的两个string数据 move 和move1：

Move用于接收输入的数据，move1为对move进行判断后返回的数据类型；

1. Boolean player 用于判断左右；

4. 主要函数及其作用：

（1） createTileAndAnimal: 用于对atlas进行赋值；

（2） judgemove：对输入进行判断；

（3） 用于打印的方法：printAtlas: 用于打印Atlas；

PrintPromt： 用于打印指令介绍；

PrintPromt2： 用于不能识别什么指令；

Printwinner: 用于打印胜利者以及胜利类型；

（4） 用于判断胜负类型的方法： win ： 用于判断是哪种类型的胜利，并返回约定值；

noAliveAnimal: 用于判断是否还有动物还活着；

beStuck: 用于判断是否所有的动物都被围住了；

（5） 用于改变棋盘的方法及辅助方法：

ChangeAtlas：首先找到那个要移动的动物，然后对应上下左右移动的类型，进行判断是否可以移动，可以就移动，改变数组；

W ,S ,A, D : 分别用于判断是否可以进行上下左右的移动，并按照相应情形（按照约定）返回相应的int ；

wTiaoShui , sTiaoShui ,aTiaoShui , dTiaoShui : 主要针对老虎和狮子，先判断是否在水边，然后再判断是否可以移动；

isShui : 主要用于辅助老鼠的下水；判断该位置是否为水域，如果是，移动之后要补上相应的水；

isXian ： 用于判断这个地方是否为陷阱，如果是，离开是要补上一块陷阱；如果是己方陷阱，则可以随意移动；

canMove : 将当前的字符串，和将要到的那个位置的字符串进行比较，判断是否能够进行移动；

1. 其他 ： whatAnimal : 用于判断是哪一种动物；

copyArray用于复制数组；

5. 优点：（1） 每一种情形都非常清晰，出了bug 易于查找；

（2） 代码的重用率还可以，再进行beStuck 的判断时，只需要用三十行就可以搞定了；

（3）对输入进行了判断；

（4）对每一种胜负情形进行了判断并且返回了相应的值；

（5）还有就是每个变量的命名还算合理；

缺点： （1） 代码太冗长；尤其时某些返回int的方法，实在是长得可怕；

（2） 在解决数组输出时还是蛮麻烦的，不如一些同学的三个棋盘合并的方法，虽然在之前createAnimalAndTile 有这样的思想，但是没有应用；

（3） main方法的结构比较复杂，不容易理解；

（4） move和move1的设置非常奇葩；

6. 遇到的问题以及解决的方法：

（1） 最初因为学得比较慢，数组什么的都不懂，就没有开始；通过加速学习数组什么的，然后才终于开始；

（2） 最初觉得可以用一个大型的文档来储存所有的棋盘信息，但是不知道这是哪一类，就一直没有行动； 后来学完数组发现直接把它们放在一个更大的数组里面就好了；

（3）最初的时候不知道两个txt文档怎么用；然后通过一节lab课明白了file的读入方法；

（4）一开始的时候真是一片空白，不知从何开始；然后我就先从玩游戏开始了，然后学习了方法那一章，就开始用自顶向下的模式开始设计我的pj；

（5）自顶向下的时候发现，有很多东西没有办法细化，不知道哪些可以结合，特例如何处理；先做普通的情形，然后在将特例加入，这样的话，整体性不是很好，就是了；

（6）之后，不知道如何处理多种吃子的情况；然后就在草稿纸上写了，然后进行归纳整理，然后发现这种方法虽然很笨，但还是蛮管用的；

（7）之后，发现自己的代码太长了，要改的时候挺难改的；选中一部分，注释，用那个减号收缩，等到我所有的都改完，再进行反注释；

（8）就是要做的事情太多了，不知道先做哪一个，然后其他部分怎么处理；用todo

（9）知道了如何将地图返回之前的那一步不知道怎么多步悔棋；设置currentStep 用于计数；

（10）代码块越来越长，不知道如何更改；利用方法块进行分装；

（11）不知道如何运用子类；当然这个问题并没有解决；

7.意见与建议

（1）intellij 对于我来说还是一个技术暗箱还是有挺多不懂的地方，可不可以多教几个快捷键，可不可以多教几个debug的招式；

（2）周四周五周六可不可留下几道思考题，但是可不可以不要跟期末成绩挂钩，不然要是那几天比较紧张，那就gg了；

（3）lab课的内容可不可以多一点提示，让我们有点心理准备，乃至预习一下，笨鸟先飞嘛；

然后，就没有然后啦~~~